Theo Documentation

Release 0.8.2

Michele (macno) Azzolari

Nov 30, 2018

Setup

Theo App is a public key manager, you can use it as replacement for all of your *authorized_keys* It allows you to set fine permissions (specific user and host) or to use wildcard (ex, using host *%.test.sample.com*)

CHAPTER 1

First steps

Theo is based on 3 components:

1. theo, the core HTTP application
2. theo-cli, the command line interface to administer Theo
3. theo-agent, the program that will be executed by sshd to retrieve AuthorizedKeys

- **Getting started**: *Cookbook*

1.1 Setup

1.1.1 Cookbook

While it's possible to install **theo** and the other components on one single server, you will appreciate all the power of **theo** with multiple servers. We'll illustrate here a scenario with 2 servers (one for **theo**, the other for **theo-agent**) and a computer for **theo-cli**.

Let's assume the server on which we will install **theo** is `server` and the other is `node-a`

theo

Install with docker

On `server` you can easily run **theo** as docker container

NOTE don't forget to replace *ADMIN_TOKEN* and *CLIENT_TOKENS* values!

```
$ docker run --rm -v /tmp/theo:/data \
    -e DB_STORAGE=/data/theo.db \
    -e ADMIN_TOKEN=12345 \
```

(continues on next page)

(continued from previous page)

```
    -e CLIENT_TOKENS=abcde,fghij \
    -p 9100:9100 theoapp/theo
```

Executing the command will result in a running instance of **Theo** listening on port *9100* and accepting calls from:

- theo-agent using token *fghij* [1]
- theo-cli using token *12345* [1]

[1] Token are sent as HTTP header *Authorization: Bearer *token**

Install from sources

Please refert to *Full install* to install `theo` from sources

theo-cli

Install

To manage `theo` we use `theo-cli`

`theo-cli` is a node app, available on npm, install it on your computer

```
$ npm install -g theoapp-cli
```

`theo-cli` needs 2 variables: *THEO_URL* and *THEO_TOKEN*. You can set them as environment variables:

```
$ export THEO_URL=http://server:9100
$ export THEO_TOKEN=12345
```

Note Refer to `theo-cli` *install document* for other ways to set these variables

Create first account

Now you are ready to create the first account on `theo`

```
$ theo \
    accounts add \
    --name john.doe \
    --email john.doe@sample.com
```

Add public key to account

Now you need to add a public key to john.doe, you'll use your public key

Note if you don't have it or want to generate another one see *generate ssh key*

```
$ theo \
    keys add john.doe@sample.com \
    -k "$(cat ~/.ssh/id_rsa.pub)"
```

Add permission to account

Now we need to add permission to john.doe@sample.com to access `server` as `root` (or other existing linux user of `server`)

```
$ theo \
    add \
    --user john.doe@sample.com \
    --host node-a \
    --user root
```

theo-agent

Download

theo-agent is a program written in go. You need to connect to `node-a` and run

```
$ sudo curl -L -o /usr/sbin/theo-agent \
    https://github.com/theoapp/theo-agent/releases/download/$(curl -L -s -H 'Accept: application/json' https://github.com/theoapp/theo-agent/releases/latest |sed -e 's/.*"tag_name":"\([^"]*\)".*/\1/')/theo-agent-linux-amd64
```

And make it executable

```
$ sudo chmod 755 /usr/sbin/theo-agent
```

Install

You need to create a system user:

```
sudo useradd --comment 'Theo Agent' \
    --create-home /var/cache/theo-agent \
    --shell /bin/false
```

Configure

You can let **theo-agent** to configure itself automatically:

With this command you will: disable ssh password authentication, disable AuthorizedKeysFile from user's home (sshd will look for them in /var/cache/theo-agent/%u)

```
$ sudo theo-agent -install \
    -no-interactive \
    -sshd-config \
    -url http://server:9100 \
    -token fghij
```

Final check

Now you're ready to test if everything is working, connect from your computer to `node-a`

```
ssh root@node-a
```

Congratulations!! You made it!

1.1.2 Full install

WIP

See

- **theo** *intallation guide*
- **theo-cli** *intallation guide*
- **theo-agent** *intallation guide*

1.1.3 Generate SSH keys

To generate SSH

```
$ ssh-keygen -b 4096
```

Leaving all the defaults, the command creates a new key in `~/.ssh/id_rsa`.

The public key is `~/.ssh/id_rsa.pub`

The public key will be used by the remote server to authorize the connection.

1.2 theo installation

1.2.1 With docker

```
$ docker pull theoapp/theo
```

1.2.2 From sources

Clone repo

```
$ git clone https://github.com/theoapp/theo-node.git
```

Install dependencies

```
$ npm i --no-optional
```

Build

```
$ npm run build
```

Configure

To configure there are 2 ways:

1. Using environment variables
2. Using `settings.json`

1. Environment variables

Name	Mandatory	Default value	Meaning	Type
PORT	NO	9100	The port on wichh the http server will listen	int
DB_ENGINE	NO	sqlite	Use sqlite3 as database. Needs DB_STORAGE Use mariadb as database. Needs DB_HOST, DB_USER, DB_PASSWORD, DB_NAME	enum: * sqlite * mariadb
DB_STORAGE	NO	./data/theo.db	Path to sqlite3 db. Absolute or relative to node process	string
DB_HOST	YES(1)		Mariadb server hostname or ip	string
DB_USER	YES(1)		Mariadb username	string
DB_PASSWORD	YES(1)		Mariadb password	string
DB_NAME	YES(1)		Mariadb database	string
ADMIN_TOKEN	YES(2)		Admin token	string
CLIENT_TOKENS	YES(2)		Client tokens: comma separated	string
CORE_TOKEN	NO		Core token	string
CACHE_ENABLED	NO		if set, the cache server will be used	enum: * redis * memcached
CACHE_URI	YES(3)	localhost:11211 redis://localhost:6379	memcached connection url redis connection url	string
CACHE_OPTIONS	NO		Optional cache parameters	string

(1) Mandatory if DB_ENGINE=mariadb

(2) Mandatory if CORE_TOKEN is not set

(3) Mandatory if CACHE_ENABLED=memcached or CACHE_ENABLED=redis

NOTE It's possible to save the variables in a *.env* file in the project's root

2. settings.json

It possibile to use `settings.json` file in the project's root to load **theo** configuration.

```
{
  "admin": {
    "token": "ch4ng3Me"
  },
  "client": {
    "tokens": [
      "njknsjd2412fnjkasnj",
      "knkjnknfjfnjenkln"
    ]
  },
  "sqlite": {
    "path": "./data/theo.db"
  },
  "server": {
    "http_port": 8890
  },
  "cache": {
    "type": "memcached",
    "settings": {
      "uri": "localhost:11211",
      "options": false
    }
  }
}
```

Run

```
$ npm start
```

1.3 theo-cli installation

1.3.1 NPM

```
$ npm i -g theoapp-cli
```

1.3.2 Sources

Clone repo

```
$ git clone https://github.com/theoapp/theo-cli.git
```

Install dependencies

```
$ npm install
```

Build

```
$ npm run build
```

1.3.3 Configuration

theo-cli needs 2 variables to work: *THEO_URL* and *THEO_TOKEN*. They can be set as environment variables:

```
THEO_URL=https://your.server.name THEO_TOKEN=your_secret_admin_token theo accounts␣
↪list
```

Or they can be stored in a file:

```
THEO_URL=https://your.server.name
THEO_TOKEN=your_secret_admin_token
```

theo-cli will look at (in this order):

```
$PWD/.env
$HOME/.theo-cli/env
/etc/theo-cli/env
```

1.4 Signed SSH keys

WIP

1.5 theo-agent installation

1.5.1 Download

1. Simply download one of the binaries for your system:

```
# Linux x86-64
sudo curl -L -o /usr/sbin/theo-agent https://github.com/theoapp/theo-agent/releases/
↪download/$(curl -L -s -H 'Accept: application/json' https://github.com/theoapp/theo-
↪agent/releases/latest |sed -e 's/.*"tag_name":"\([^"]*\)".*/\1/')/theo-agent-linux-
↪amd64

# Linux arm
sudo curl -L -o /usr/sbin/theo-agent https://github.com/theoapp/theo-agent/releases/
↪download/$(curl -L -s -H 'Accept: application/json' https://github.com/theoapp/theo-
↪agent/releases/latest |sed -e 's/.*"tag_name":"\([^"]*\)".*/\1/')/theo-agent-linux-
↪arm
```

2. Make it executable

```
sudo chmod 755 /usr/sbin/theo-agent
```

3. Create a Theo Agent user:

```
sudo useradd \
    --comment 'Theo Agent' \
    --shell /bin/false
```

4. Install

4.1. Full Automatic install

```
sudo theo-agent -install \
    -no-interactive \
    -sshd-config \
    -url ${THEO_URL} \
    -token ${THEO_CLIENT_TOKEN}
```

4.2. Semi-Automatic install

```
sudo theo-agent -install \
    -no-interactive \
    -url ${THEO_URL} \
    -token ${THEO_CLIENT_TOKEN}
```

Edit `/etc/ssh/sshd_config` as suggested

4.3. Semi-manual install

```
sudo theo-agent -install
```

Answer to the questions and edit `/etc/ssh/sshd_config` as suggested

4.4. Manual install

Create a `config.yml` file (default is */etc/theo-agent/config.yml*):

```
url: THEO_URL
token: THEO_CLIENT_TOKEN
```

Create a cache directory (default is */var/cache/theo-agent*):

```
mkdir /var/cache/theo-agent
chmod 755 /var/cache/theo-agent
```

Modify */etc/ssh/sshd_config* (if you changed the default path, add the options to the command)

```
PasswordAuthentication no
AuthorizedKeysFile /var/cache/theo-agent/%u
AuthorizedKeysCommand /usr/sbin/theo-agent [-config-file /path/to/config.
↪yml] [-cache-path /path/to/cache/dir]
AuthorizedKeysCommandUser theo-agent
```

CHAPTER 2

Indices and tables

- genindex
- modindex
- search